


Graduado en Ingeniería Informática

**Universidad Politécnica de Madrid**

**Facultad de Informática**

**TRABAJO FIN DE GRADO**

# Participatory Design and Business Modeling

## Shared Interests

Pedro Iglesias De La Vega

Supervisor: Troels Sune Mønsted


MADRID, JULIO DE 2014

# Table of Contents

# 1. Resumen en Español (Summary in Spanish)

En este proyecto miraremos como los campos de Diseño Participativo de las tecnologías de la información (DP, Participatory IT Design) y el diseño empresarial de modelos de negocio interactúan entre sí. Exploraremos el hueco que existe en los asuntos que aborda un proyecto de DP señalado por Kyng (2010).

DP es una forma de enfocar el diseño para diseñar sistemas informáticos que tiene su propia ideología, haciendo mucho hincapié en la participación de los usuarios en el proceso. Esta ideología propia de DP da paso a un conjunto de métodos y herramientas para diseñar. DP tiene sus orígenes en Escandinavia en la década de los setenta cuando cambios de progreso social de la sociedad se introdujeron en el lugar de trabajo. Ya hay algunos investigadores del campo de DP que empiezan a estudiar este hueco señalado por Kyng.

Un modelo de negocios es la forma en que un negocio crea, entrega y captura valor. En las últimas décadas las empresas informáticas han cambiado su modelo de negocios, transformándose en una oferta de servicios informáticos mayoritariamente. El mundo empresarial ya conoce los beneficios de las metodologías de diseño orientadas al usuario, sin embargo no acaban de ser adoptadas por las empresas informáticas.

En el proyecto buscamos responder a las preguntas que surgen al juntar estas dos áreas de estudio. ¿Cómo interpreta la comunidad de DP el mundo empresarial? ¿La teoría empresarial como trata la metodología de diseño orientado al usuario? ¿Cómo afecta en una compañía real su política de usuario a su modelo de negocio y que técnicas utilizan para involucrar al usuario? ¿Qué intereses comunes y opuestos existen entre DP y el mundo empresarial? ¿Cuál es el futuro de DP en el mundo empresarial?

En cuanto a la metodología del proyecto realice un estudio de la literatura de los campos de DP y el mundo empresarial usando fuentes recomendadas por expertos de cada área así como trabajos encontrados por mi propia cuenta que eran relevantes al tema tratado. Además realice un estudio de compañías desarrolladoras de software locales a Copenhague. Mediante una serie de visitas y entrevistas con esas compañías obtuve información del modelo de negocio y proceso de desarrollo de software de las mismas. También dialogamos en cuanto a su visión de DP y metodologías de diseño orientado al usuario y como se podrían beneficiar de ellas.

Durante el desarrollo del proyecto he encontrado que DP y el mundo empresarial tienen dos visiones muy diferentes. Mientras uno busca obtener el máximo beneficio el otro busca apoderar a los usuarios en su área de trabajo. Aunque al igual que DP las empresas necesitan obtener información acerca de los usuario como sus necesidades, gustos, etc. Está claro que el software producido para el mundo empresarial tiene mucho espacio para mejorar ya que suele venir con manuales de instrucciones extensos y es necesario un entrenamiento previo para su utilización. Un problema importante de la introducción de DP en las empresas es la falta de predisposición de los usuarios en colaborar en el proceso de desarrollo ya que no hay demasiada tendencia a usuarios a

dar feedback. El trabajo de investigacion de Timpka y Vimarlund (1998) *Participatory Design In Economic Terms: A Theoretical Discussion* llega a la conclusión de que DP es económicamente favorable para los desarrolladores. Pero omite muchos factores críticos de la economía de una empresa.

En cuanto al futuro de DP en el mundo empresarial, DP tiene que adaptarse a las compañías desarrolladoras de software. A sus necesidades y teniendo en cuenta sus límites de recursos. Pero hay ciertos aspectos que deben ser abordados. El primer aspecto es el uso de recursos en las técnicas de DP, ya que utilizan demasiados. Otro aspecto que debería abordar es el uso de las nuevas tecnologías de comunicación y redes sociales, ya que apenas se han utilizado en DP. Mientras que otras áreas de estudio han hecho uso de estas nuevas tecnologías y se han revolucionado DP no ha sido capaz de integrarlas en sus técnicas.

Por último en el proyecto realizo una comparativa del actual modelo de negocios y proceso de desarrollo de software de la compañía Forecast.it. En esta comparativa miro como cambiaria el negocio en el caso de que se produjeran cambios en ella al introducir la ideología de DP. Donde más se notan los cambios seria en su esquema de desarrollo de software ya que habría que hacer hueco para nuevos factores y pasos.

# 2. Introduction

Since the establishment of Participatory Design it has tried to expand in order to try to benefit as many people as possible. And, as a way to reach the most number of people, the business world is ideal as it produces the majority of software out there. The interfusion of these two fields touches a fundamental issue for the future of Participatory Design and its community. Participatory Design positions itself as a relevant aspect for a business as it is a tool to produce better and innovative products for businesses to commercialize. Software development business are told that they need user involvement because they are told by the market that they need to produce better software products. So these two fields seem like they form an obvious benefitting partnership. Yet the reality is that Participatory Design is not being fully adopted by software development companies. They prefer to improve their products through other means or innovate through other aspects. So to be able to see if Participatory Design can become a common tool in today´s modern software development sector we must research this area of study. We must see what shared interest are common to both Business modeling in the software developing sector and Participatory Design, as there is sure to be some common interests parting from the fact that they both wish to produce better products/software. As well as the shared interest we must research the opposing interests and what hinders the total adoption of Participatory Design by business companies. This issue has largely arisen by researchers, such as (Kyng, 2010), pointing out at a gap in Participatory Design study where the politics of a design project are largely ignored. And in this gap lies the adoption of Participatory Design by businesses.

In this project we will study this largely unexplored gap pointed out by Kyng, to know how these two fields of study, Participatory Design and business modeling, overlap, as Participatory Design tries to reach towards a business environment. I will find out how PD interacts with the project reality in a business environment. To see if they clash or they cooperate in favor of joint interests.

# 3. Background

## 3.1.Participatory Design

Participatory Design (PD) is defined in Handbook of Participatory Design as " a process of investigating, understanding, reflecting upon, establishing, developing and supporting mutual learning between multiple participants in collective 'reflection-in-action'. The participants typically undertake the two principal roles of users and designers where designers strive to learn the realities of the users' situation while the users strive to articulate their desired aims and learn appropriate technological means to obtain them" (Routledge International Handbook of Participatory Design, p. 2 ). So PD is an approach to IT design with an ideology. This approach gives way to a set of methods in order to obtain a design which proves most useful for the user and thus empower him. These methods consist of incorporating the workplace practitioner into the design process so he is the one who ultimately moulds the system. The end user is

invited as the expert in their field of work to the design process. It is a methodology that tries to empower the end user in the workplace. This sets it apart from other user-centered design methods such as contextual design which merely have the needs and preferences of the end user in their sight but do not involve him in the design process as an equal expert(Contextual Design: Defining Customer-Centered Systems, Ch. 1). Users do not always know what they want and thus might not be able to properly explain it. There is also generally a huge difference between the ideal method for working and the manifested working behavior. As thus a user in an interview might describe his practices in a way that does not relate entirely to the reality.

Participatory Design has its roots in Scandinavia in the 1970´s, when changes occurred to the society at large, rooted in local communities as well as workplaces; it was a time when progressive ideas from the society started to spread into the computer systems design world. At this moment computer systems were being introduced into the workplace. PD came to be as workers teamed up with computer scientist in order to democratically bargain how computers should be introduced into the workplace and thereby empower the workers (Routledge International Handbook of Participatory Design, Ch. 2 ). The origin of PD reflects a policy that commits to ensure that the opinions and voices of marginalized groups and communities are heard and evaluated in the decision making processes that will have an impact on them. The participation of the users happens because those who will be affected by the new information and communication technologies must have a chance to partake in the process of their design and influence its final form and the practices of its use. In PD users are viewed as a project partner rather than an object to be studied.

Participatory Design becomes a contrast to traditional methods of IT design. These methods are rooted in the origin of computer science of engineering and natural sciences. They focus on implementing clearly detailed specifications, assuming these exist, but the reality of the practice is very far off. Step by step procedures of traditional methods prevent a creativity and a cooperative relationship between the designers and the users(Design at work: Cooperative Design of Computer Systems (1991), p. viii). Thus hinging the overall quality and usability of the final design. At the time when PD began many experts realized that when those who would use the new computer-based systems were not actively involved and influential in their development and use, it was not possible to create  future conditions and practices that would improve or even match the current ones. So PD focuses on the user and has him design the system thus ensuring that the system is designed in accordance to his needs and his working conditions.

Since its origin PD has established an extensive worldwide community around its practices, history and ideals. This community hosts a biannual conference, named Participatory Design Conference, in different cities around the world to exchange ideas, methods, discoveries and the likes pertaining to PD. Currently the participatory design community is in a time when many experts in it believe that a change must occur in order to further develop PD and insert it into common practices of information systems design. This change must come in the PD community addressing the politics in a PD

project which are normally ignored, such as funding, the role of companies or intellectual property rights (Kyng, M., 2010). With these aspects of a project acknowledged the implementation of PD in more diverse context can become a reality and thus extend its use in common day practices. Modern day PD projects "often [end] before we can support users to analyse and resolve issues that might only become visible once the system is in use" (Routledge International Handbook of Participatory Design, p. 79 ) or are "small, local and isolated initiatives with small impact outside the project" (Routledge International Handbook of Participatory Design, p. 131 ). By addressing the project politics the number of these types of situations can be reduced greatly and thus further develop PD ideals.

Kyng (2010) also suggests that the emphasis of PD projects in the community have shifted somewhat, in the sense that researchers have as a higher priority to produce research papers than to produce teaching materials. Ellen Balka (2010) writes in her response to Kyng that this would partly be due to the change in academia in general in the increase emphasis of research outputs in the form of research papers. This could also be due to the fact that sponsors of such projects expect papers to be written in order to have something to show for the capital invested. So here lies another example of issues regarding politics surrounding the implementation of PD projects and how they are relevant to the community.

As a way to address the issue of funding in PD projects some researchers have very timidly reached out on the topic of external funding in the private sector. These few researchers have realized that in order for PD to increase the number of people it touches it needs to broaden what it encompasses to include the politics of funding for the projects, and how to carry out a PD project in a business environment. Already one of the most recent methods of PD, the MUST method, is a business oriented approach (Participatory IT Design: Designing for Business and Workplace Realities, Ch. Introduction). So this method is already tried to adapt significantly to modern day business environments. The problem is that although it tries to adopt business reality into PD the business reality does not adapt to it.

Other studies such as Toni Robertson´s paper on "Participatory Design and Participative Practices in Small Companies" have tried to expand the practices of PD to not only include larger organizations in their projects but also small companies. As Robertson (1996) puts it "Participatory Design´s traditional focus on large workplaces risks its self-definition as relevant to only limited domains of work". So there were hints about a longing in the community to expand its borders to all domains possible antecedently. Other researchers have made their contributions to this aspect with different papers. Vivian Vimarlund and Toomas Timka describe PD in an economic perspective in their research paper "Participatory Design In Economic Terms" (1998). They suggest that " shared investments during [PD] processes are beneficial for the organizations involved by an increase in knowledge capital, a decrease in information asymmetry due to the different backgrounds of the participants, and by updating work routines in a 'natural' manner". This study, however, is purely theoretical and does not contain empirical

data; still this in theory makes it a viable option for businesses of all sizes to implement PD methods instead of traditional methods. This gives way to a more enriched product or service, and thus also further gives value to the company and accomplish the task of extending PD ideals such as empowering the user. With this study we know that it is a competitive option for business owners to implant PD methodology into their business model.

The notion of seeing how PD interacts with business model has been taken by others in completely different directions. Jacob Buur tries to apply ideals and methods from PD to the area of business modeling in his workshop conducted in the 2012 Participatory Design Conference (Participatory Design of Business Models, 2012). Although it is not Participatory IT Design it does demonstrate that the PD community does see the potential of expanding its ideals in the business world and tries to reach towards it.

## 3.2. Business Modeling

"A business model describes the rationale of how an organization creates, delivers, and captures value" (Osterwald, A and Pigneur, Y., 2009) . So a business model of a company is basically the way that company does business. It encompasses all aspects of the business, from what key partners it has to its different revenue streams. Osterwald and Pigneur (2009) divide a business model into 9 basic building blocks: customer segments, value propositions, channels, customer relationships, revenue streams, key resources, key activities, key partnerships and cost structure. To define these blocks is to define the business model of a company. A success of a company depends largely on its business model. Two companies producing the exact same product or service might not have the same success if one has a successful business model and the other has an outdated and inefficient one.

Taking a look at the 9 basic building blocks you will see there is a large emphasis in the customer. The business model includes to which customer segment the business is directed to, what value it creates for the customer, through what channels the value is delivered to the customer and customer relationship. So just like in PD there is also a large presence of the customer/user even if it is not with the same policy behind it.

There has been a huge change in how software companies do their business now compared with the beginning of the IT industry in the 1970´s. In the beginning software companies were prone to offer the sale of individual products, but the changes occurring now transform this offer from an individual product to a regular service which is paid by a subscription fee (M. Cusumano, 2008). So this shift in business model types consequently alters many aspects of the sector. Specially the type of customer relationship, and the channels by which it receives the value it needs. These changes revolutionize how people fulfill their IT needs. They interact with the companies in completely different manner expecting different things then what was originally offered in the beginnings of the sector. These changes makes the importance of having an updated business model specially significant for the IT sector.

Thus in the software industry there is an increasing awareness of the importance of business modeling. This importance can be seen by the increased research into the business model world by the IT community. Markus Schief highlights this importance in his recent book "Business Models in the Software Industry: The Impact on Firm and M&A Performance" (2014). This awareness of the importance is a consequence of the hight competition in the sector as there are many IT companies out there developing the same products/service and they all wish to gain the upper hand over their competitors. So they know the difference between success and failure lies largely in the business model.

To be able to gain the edge over their competitors many companies have realized the usefulness of user-driven design methods of their products or services. Companies like LEGO allow users to design their own LEGO sets, and then the LEGO factory sells those user designed LEGO sets online. Many businesses have seen the advantage of having users design new innovative product (Osterwald, A and Pigneur, Y., 2009, p. 29). Through business modeling user-driven design methods have been known to few numbers but many different types of companies. They see that they can create unique value through the customization of a product through user-driven design methods. But software development companies have rarely seen this as a viable business oriented option.

# 4. Problem

In this project and its subsequent final report I will touch various problems of how PD overlaps with the field of business modeling, and how these two fields interact with each other when they interweave. Problems have arisen in the PD community with the successful introduction of PD into a business model. This interaction between the two fields lacks research publications in many of its aspects, as it is pointed out earlier in the background section.

I will study what is currently said about business models, and the business environment in the PD community and its literature. How does the PD community interpret the business world? Does it see it as an opposite of its democratic philosophy and its goal to generate a democratic workplace? Does it see it as a way to be useful to the most number of workers? These questions are partially answered in the background section, but it will be explained in more detailed the stand taken by the PD community and its literature.

I will also study what the business world and its literature says about user-driven design methods and PD if anything is said about it. As PD is a very narrow field and it is more common to talk about user driven design methods than PD, I will look more for how they interpret user driven design methods and their policy on involving users. How open is the business world to user driven design methods on paper? Is there a general consensus to how to generate and deliver value through these methods? I will see what

is written down in the business world and academia about how to generate value through user-driven design methods.

Through the company study I will be able to solve problems related with how they actually approach user involvement in their companies. How does their policy on user involvement affect their business model? What type of techniques do they use to involve users and see what they require? Can any particular aspect of their business benefit from PD? I will see with real data what real companies do and their approach to user involvement and user driven design methods to better understand what current practices are and the trend in the sector.

Through the answering of the previous questions and through the combination of the knowledge gained through the company study and literature study I will find out what common and opposed interest there are between PD and business modeling. Do they have any common ground where there interest coincide? Is there any opposing factors between these two fields? If there is an economic factor in favor of PD as pointed out earlier, what is hindering the introduction and consolidation of PD in the business realities? These questions are the basis of the project, and the most in-depth questions I will answer.

Once answered the previous questions I will be attempt to make a prediction and recommendation of how we can introduce PD into a business model. How can we create value for a company through PD? What aspect of the business model can PD enhance and make the company benefit? The goal is to able to make realistic and useful recommendations of how to improve a software development business through the use of PD and how to introduce it into a modern business model. We will attempt to adapt one of the business models and practices from a company in the company study in order to fit in it PD. So to be able to do this we will need the knowledge of the previous stated questions. Through the responding of the questions stated in this chapter I intend to meet the goals of the project and thus explore the boundary of PD and business modeling.

# 5. Methodology

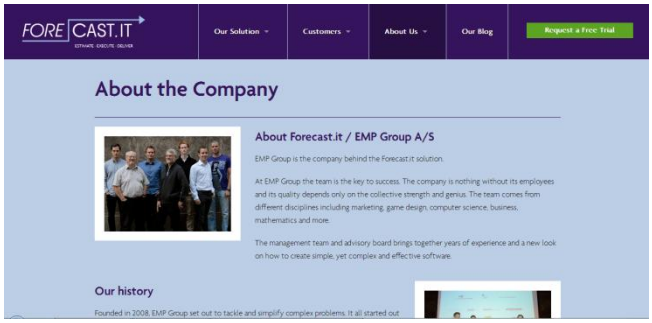
In this section I will detail the methods used to carry out this project. In order to try to resolve the problems and questions stated previously I will use two methods: Literature review and a partial company study. Each method will be carried out separately although they will overlap time wise. The selection of methods comes from a need to gain as much knowledge as possible from the business world and the PD community having a very limited access to both fields. As such I needed to make the most out of the very little time companies could afford to lend as well as review as much relevant published knowledge of both fields as I could.

- **Literature Review**: This method is *based* out of Harris M. Cooper´s (1998) *Synthesizing Research: A Guide for Literature Reviews* although I did not follow

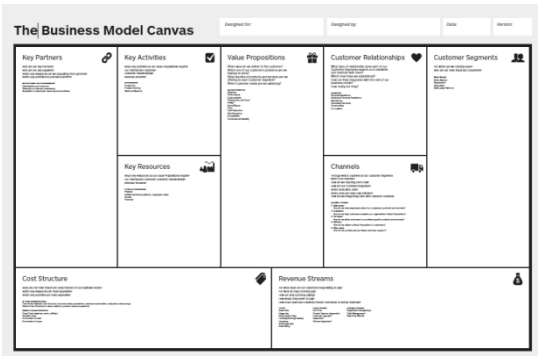
it completely. Literature review has a lot of different meanings but what all definitions have in common is that they are research "not based primarily on new facts and findings, but on publications containing such primary information, whereby the latter is digested, sifted, classified, simplified, and synthesized" (Manten 1973, 75). My literature review consisted in the systematic reading of different research papers and books and highlighting ideas that seemed relevant to my project or interesting for me. Once compiled all these ideas I proceeded to analyze them and attempt to produce new ideas from them. With this method I will aim to obtain a general and in-depth idea about both PD and business modeling. The review will also focus on what is written in the PD community about the business world and business modeling and what the business literature says about PD or user-driven design methods. To carry out this review I will select the most generally widespread literature works in each field as well as less known literature which might address the issues of the opposite field. To find this literature needed for the project I have varying methods depending on the field. For PD I have scouted the PDC proceedings and Scandinavian Journal of Information Systems for papers which seem relevant to the project, as well as broad well known PD books such as *Routledge International Handbook of Participatory Design* and *Participatory IT Design. Designing for businesses and Workplace Realities* and some other minor research papers I came across through the use of Google scholar. I looked for papers which somehow touched the world of business model, either directly or indirectly. For the field of business modeling I used the now very famous *Business Model Generation: A Handbook for visionaires, Game Changers and Challengers,* which some experts in PD have already cited, as well as other business related sources that PD experts reference. Lastly I also search Google scholar for minor paper concerning business modeling of software development companies.

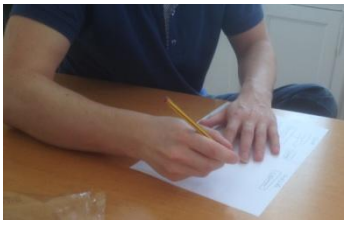
- **Partial company Study**: This series of methods is based on *Workplace Studies: Recovering Work Practice and Informing System Design,* where they define it as "a corpus of studies [...] that [is] concerned with revealing the details of how activities are accomplished in real-world workplaces," (2000) as well as methods to gain knowledge of current practices from the MUST method from *Participatory IT Design: Designing for Business and Workplace Realities* as I have previous experience carrying out these techniques.. As I have previous experience in the methods of MUST these were my main inspiration and guide as I did not do any field work per se. With this company study I will intend to learn the current practices of modern software development companies as well as their business models. I will also try to assess how PD could help the company being studied and share the outcome with the company to see how it would fit into their business model. In this company study I will work with two software development companies situated in Copenhagen: Forecast.it and Gluu. In order to undertake this company study I used the following techniques :

- *Preliminary Study*: This technique is based on the MUST method technique of Document analysis (Participatory IT Design: Designing for Business and Workplace Realities, p. 208). With this technique I intend to gain a first insight into the company, its structure, methods, products, what they do and business model. I do this by examining its webpage and gaining as much insight as possible from all the information placed online. I study the aim of the company, what products/services it offers, employees and function of each employee.




- First Interview: This technique is based on the MUST method techniques of in situ interview and workshop(Participatory IT Design: Designing for Business and Workplace Realities, pp. 207-211).With this technique I will gain the majority of knowledge from the company. It will consist of an interview/workshop to a CEO of the company and a software developer. The interview/workshop would consist of:
  - *Completion of Business model Canvas*: With this tool I ask the participant to fill out with me the business model canvas (Osterwald, A and Pigneur, Y., 2009) while he narrates how his business works and current practices. I use the Business model canvas not as it is intended in order to design a business model, but in order to start the discussion and prompt their reflection in how their company works. Through this step I expect to gain an insight into the business model of the company and see how their business is run.



  - *Software development scheming*: Here I ask the software developer to draw a diagram of the general process of developing software. This way I gain an insight of what the process consist of. I find out where they get the information of requirements, how do they design the user interface, etc.
  - *Question and answer*: In this section the participant is asked about any other aspects that might be unclear about their business model or development process. I also ask different questions such as their user policy used, how is their client satisfaction with their

software, if they have had any mentionable failures or successes, etc.

- Second interview: This interview as with the first is based on the MUST method technique in situ interview (Participatory IT Design: Designing for Business and Workplace Realities, p. 207) In this Second interview with the company, in which only Forecast.it participated, I will focus on the introduction of PD in their business. After explaining what PD is I will ask the CEO how PD can result beneficial to their business in anyway. I will as well propose ideas about how to introduce PD in the company and make them benefit from it. The company will then give me feedback on these proposals.

# 6. Findings & Discussion

In this chapter I will detail what I have found in relation to the literature review and company studies as well as discuss the findings. I will start from the perspective of PD to see how it encounters business modeling and business reality, follow it by situating myself in the business reality and seeing how they look at PD and user involvement. I will continue with how these two fields come together and what can be done about it. And finally I will produce an example from the company Forecast.it as to how a company may use PD and how it would affect its business model and how it produces software.

## 6.1.Participatory Design

### 6.1.1. Visions

The business world has a distinct vision than the PD community as to the way they look at users. For PD, as has already been stated, they are the goal to reach the users and empowerment, they are the end and PD is the means (Routledge International Handbook of Participatory Design, Ch. 1 ). While for the business reality traditionally clients are a means to the goal which in this case is to obtain a profit( Core Sociological Dichotomies, p. 383, Business Model Generation: A Handbook for visionaires, Game Changers and Challengers) although of course there are other types of businesses which are not entirely set on profit but have other goals, but I will concentrate on these more traditional capitalism values of gaining wealth. So there is a clear difference in perspectives, while PD would gladly sacrifice low costs to obtain better usability and more adaptation to the user, a company would not venture into such territory if it was not sure it would make a profit or have a good chance at making it. There is however common criteria, where both realities share goals.

### 6.1.2. Participatory Design in companies

Software development companies have been shown that do need to know what and the reason for what a client needs in order to better produce software (Business Model Generation: A Handbook for visionaires, Game Changers and Challengers, Ch. 1). In the case of Forecast.it during their negotiations of a proposal they sometimes would get

requests for a certain feature, and when asked why the developers would notice that the feature would not work or would be better done in another manner and thus propose a better solution to their problem(Appendix A). This shows that there is a real need in actual companies to find out what the users need. As finding out the current practices of a user is an essential part of any PD methodology, PD can benefit companies and introduce some of its methodology which contain its ideals. Even if it does not mean the complete user involvement in the design process. Obviously you cannot speak of PD without user participation in the design process as it is in its very definition, but PD has developed very thorough and reliable techniques and tools to obtain current practices of the users such as MUST or STEPS (Routledge International Handbook of Participatory Design, ch. 6-7). So companies could adopt these methods in order to understand their clients or users, this would very lightly introduce the ideals of PD of empowering the user. So traditional PD methods such as the MUST method which have its first phases initiation phase, in-line analysis phase and in-depth analysis phase put a large emphasis on learning the current practices (Participatory IT Design: Designing for Business and Workplace Realities, pp. 91-165), would be ideal for companies to use in order to meet their goals.

Through my Company study I found that software development companies who develop software for enterprises frequently have to offer a service of training for their software to their clients and have to include an extensive manual on how to use their software (Appendix A). This training service they offer in reality is considered a loss for them, as they would be able to generate a higher amount of profit if the resources utilized in their training service were committed to the further development of software or others tasks such as consulting. So the sector follows a requirement specification and delivery pattern almost, where the client specifies what features he requires and the company delivers what he asks with a follow up on any upgrades or specifics that might need changing as they follow an SCRUM methodology. As such this reveals that there is a very large room for improvement in communication between the designers and the user. So PD could prove useful to companies in this aspect of their business model where the communication with client is desired in any software development company.

Another peculiarity I found through my Company study, is that when given the chance to voluntarily express their needs, and participate in the design process , users would not always take the opportunity to express themselves for reasons such as probably times constraints from their part. In this case the company Gluu offered a platform in order to give their users a chance to give feedback such as criticism, needs or wants, and suggestions. But that platform was being ignored by the majority of users of the system (Appendix A). This opens a whole other point of discussion where it might not be possible to partake in PD not because of resource constraints or unwillingness of the company, but because there is a lack of genuine interest from the users to involve themselves in the design process. This discussion has already been mentioned in the PD community in papers such as *Obstacles To Participatory Design In Large Product Development Organizations* from J. Grudin (1993). Naturally a PD project has to

address the issue of gaining users to have as designers, so a company´s project which utilizes user involvement should as well. If it is taken into consideration the position of the users then, why should they voluntarily offer their free time to improve something they are paying for? In this case they are the clients that pay for a quality service or product and expect to get what they are paying for without further inconveniences. So if a company wishes to utilize PD methods it must account for the cost of contracting a regular user in order to work alongside designers. It is the company responsibility to obtain a quality software that satisfies the client's needs and wants and set a price on it. So in order to get the quality desired for their products or services a company must find strategies to obtain it. These strategies could include PD as a corner stone. In most PD projects carried out until today users are compelled to take part in the project helping out designers as their superiors in the client company or firm place them in that position. But in the business reality clients, be it individual clients or companies, do not wish to redirect man power from tasks of the company into helping the designers create a better quality product.

It must be discussed that there are other sectors where user driven design methods are very successful (Business Model Generation: A Handbook for visionaires, Game Changers and Challengers, 2009). This could serve the PD community as an inspiration and sign of optimism that the value of the user is being appreciated in the business reality. So PD should learn from these sectors on how to apply user driven design methods to a successful business. Through my Company study I found that Forecast.it considers a key point in their success is that their software their intuitiveness and user friendliness compared with other enterprise software in the market (Appendix A). So apart from other sectors recognizing user driven design methods, the software development sector recognizes the importance of user friendliness and intuitiveness in all types of software. The path seems already set for the PD community to take advantages of these circumstances and become more common practice in the business world if it is able to adapt.

## 6.2. Business reality

### 6.2.1. Participatory Design in economic terms

As has been mentioned in the background chapter, the study Participatory Design In Economic Terms: A Theoretical Discussion (1998) theoretically demonstrates that PD is economically advantageous for a company to undertake. This is somehow the PD community complaining about how superb PD is and how businesses have not taken an interest to it and ignored it. This study suggest that it is economically viable for business for various reasons including: lowering risk of inappropriate software, intellectual capital gain for client and designers or knowledge asymmetry that might give way to having to make further upgrades in the future. And indeed this claim is also backed up by Molokken-Ostvold and Furulund (2007) who studied 18 development projects and found that daily communication between the developers and the customers led to fewer effort overruns. But the authors of the 1998 study do not realize that in the business reality the sunk costs of producing inappropriate software after having a exchanged

needs and requirements in a meeting is practically zero as seen in the interview with Forecast.it (Appendix A).

This means that there are other more resource friendly methods which the software development industries already uses to gain knowledge of the client/user. The points made in the paper (Participatory Design In Economic Terms: A Theoretical Discussion, 1998) concerning knowledge asymmetry and intellectual capital are valid but the gain for real business pales in comparison with the cost of utilizing PD in its fullness. The knowledge asymmetry issue has gained attention in the business world as it has already been stated above that companies seek to inform clients about their capabilities and gain as much insight into the client's needs and current practices. As it is a serious issue for companies to know the needs of a company to better deliver value. The intellectual capital gain the referenced in the paper implies that software implemented through PD could lower training cost for new workers incorporating in the client company. This could very well be the case but companies generally would not want to spend upfront and invest in a measure that would not save them capital until there is a renewal of the workforce. And as studies show it is profitable for companies to delay the payment as maximum as possible if no interest apply (Economic Order Quantity under Conditions of Permissible Delay in Payment, 1985). This of course makes calculations change as the client would rather not pay for a more user-friendly, and intuitive software when it can save costs and invest in operations that have a more profitable return of investments for their company.

It all comes down to the extensive use of different types of resources the practice of PD needs. Most widely recognized PD methods just require too many resources. The four methods referenced by the now famous Routledge Handbook of Participatory Design (2012): STEPS, MUST, CESD Use-oriented Design all rely heavily on an extensive direct communication with the user, straining the use of resources in both the designers and the users. No one puts in doubt the benefit of PD for the software development sector but business already know the benefits of user involvement, as the trend is to make all software more user friendly and intuitive. But in order to achieve their goals other methods and study areas in Human-computer interaction (HCI) have already made great advances in the recent years as seen by companies such as Apple producing guidelines for app developers to keep apps user friendly and intuitive in their systems (iOS Human Interface Guidelines, Apple Inc).

So as PD is accustomed to more public funded project where generally the availability of resources is greater with an emphasis on the demand for quality products without having to excessively worry about return of investment. As already stated in the background most PD projects do not concern with the funding aspects and as such methodology normally ignores this aspect, assuming we have almost limitless access to users, designers time and other resources. But in the business reality these things are of the upmost importance. So there is a clear clash of interest in this aspect of traditional PD and the business reality. So PD is not even thought as an option for medium to small

sized companies and startups as their time and resource availability is even more limited.

### 6.2.2. Types of companies

But there is still a sector of business who could potentially benefit greatly from PD. This sector of software development is the companies specialized in customized software product or service for a specific sector. As pointed out by Osterwalder and Pigneur "Tailoring products and services to the specific needs of individual customers or Customer segment creates value" (Business Model Generation: A Handbook for visionaires, Game Changers and Challengers, p. 29). What this suggests us is that a business can create value for their clients and gain revenue through customized products or services. These customized product or services could effectively be developed through the use of PD even at its high resource costs. So a successful company can emerge where they develop high quality customized software for specific client that utilizes PD. Another possible scenario where PD would currently prove useful in the business world is for companies whose business model revolve around customer relationships. This type of customer relationship business sector follow an unbundled pattern business model where customers typically have a long term relationship with the company, customers have a high cost of acquisition and it is crucial to gain high share of wallet from each customer (Business Model Generation: A Handbook for visionaires, Game Changers and Challengers, pp. 57-59). These businesses have a business model that revolve around the sole of a healthy and well founded customer relationship. Since most modern day companies need some type of software to communicate with clients any software developed for companies in this sector would benefit greatly from PD. These companies would be willing to invest an extra so that the software that their clients, who will be the users, have the best experience when utilizing it. As here the companies interest is the favorable communication and satisfying customer experience of the client it overlaps someway with PD´s goal of empowering the user of a IT system. But this again is a very narrow portion of the whole industry.

Smaller companies and start-ups in more widespread sectors have fewer resources and thus do not even contemplate the use of PD unless a big part of their business idea, business model and whole policies revolve around user involvement which would of course make this company fall under the type previously discussed which specialize in customized software. As it is stated there has been a huge advancement in HCI and with newer generations of users more accustomed to the use of computers and having grown up with the use of all different varieties of computers, designers have much less trouble obtaining the product they desire. So users are now more 'trained' to use software through everyday lives, and designers are getting better and better at producing user friendly designs. So where does PD want to introduce itself? If the business world is capable of producing mass market software who everyone is capable of using; why would companies in the mass market sector of software jump into PD in order to produce software which would not give them a competitive edge compared to the quality software their competitors produce? If companies are able to achieve their goals

and be successful then they will not adventure into PD. Users will not donate their time in order to contribute to better software design of products or services so they must be hired as already seen. So if users who are now more technology friendly and do not want to partake in the design of the software they use, then maybe PD is not needed in these small companies and recent start-ups.

Bigger companies who sell products or services to many clients, which are a minority in reality, can afford to spend extra resources on the software they produce. This is because the cost of PD is divided into each sale they make, so if a company only sells one product which was made through PD, than that sale carries 100% of cost but they sell one million, each sale carries 0.0001% of the cost. This economic principle makes the use of PD viable for companies who commercialize to big enough customer segments. If a company spends resources on developing software of higher quality through PD and other methods to obtain a superior software then they must make enough sales of that software in order to pay for that extra cost, and be able to sell it to more clients than a small company or start up ever could.

This distinction between big and small companies although it seems logical, and my company studies confirm this as both companies stated that PD would not suit at their stage of time, it must be pointed out that most companies that use user driven design methods are considered small or are recent start-ups such as "Living IT lab" in ITU (www.sus.dk/english/) who utilize user-driven techniques. These companies as such normally have a business idea which revolves around the concept of user driven design methods, and are a minority compared to the number of software development companies out there.

## 6.3. Interaction and Future

The question of how these two fields interact arises. As it has been pointed it out in the PD section of this chapter there is a clear difference in goals, even though they have some common minor goals. So this mindset in each field will clearly not change as it is what characterizes each field. So a company cannot modify its desire to obtain profit, that much is obvious, and PD is founded on the principle of empowering the user. This is a clear clash of conflicts, as although they are not completely opposing goals, their objectives differentiate vastly. This difference in mindset makes the thought of working simultaneously with both of them in the same corporation very difficult and essentially inexistent if considered solely pure PD and not user-involvement in general. Only very specific businesses with clear goals which assimilate more closely to PD´s own goals and large enough companies with enough sales to distribute the cost could clearly benefit with today's PD methods and techniques nowadays. The first group of businesses have clients with a clear customer relationship emphasis whose most important customer relationship channels is some type of software. In this case the application of PD ideals and methodology is clear, as they have to develop very high quality and customized to a customer segment; so the use of user driven design methods like PD is ideal. But even in these sectors PD is not widely used.

So PD must adapt to each sector individually. But there a few clear things that must be addressed in order for PD to adapt to any type of sector. Firstly it must reduce resource needed for it. This includes reducing the time needed with users, and time of designers. This is the main drawback for companies not adopting PD and generally user driven design methods so a breakthrough in this field would greatly benefit its introduction. This could be done by developing new methods and techniques from within the community that put an emphasis on this aspect. As it has already been stated none of the popular methods developed so far in the PD community have an emphasis on this.

Another aspect that should be considered and addressed is the relatively low use of the telecommunication revolution that has taken place in the last two decades. Unlike most technological fields of study who have welcomed the use and been strengthen by the new telecommunication technologies such as broadband, PD has not used it to its full advantage and not many important papers about this topic have been published. In a world where telecommunications speed is the new revolution, the PD has to take advantage of it. It can take advantage of global access to the internet to strengthen its vulnerabilities like resource costs. In today´s software development industry a lot of development is externalized to other regions of the globe and thus have no possible contacts with the user apart then maybe a video call. Many companies I contacted through my company study negatively replied because their developers were situated in another city or completely different country. So PD can develop techniques which do not rely on close physical approximation and can be done from a distance. It can take advantage of the worldwide access to the internet to gain a little insight on many different users, this is especially relevant for software being developed for a mass market, as it gives access to millions of potential users. So the new technologies could do to PD what they have done to all other technological sectors and globalize it. Some papers have been publish which tackle this problem including *Distributed participatory design* (2008), *Distributed Participatory Design – A Case Study* (2006) or *Participatory Design through Social Media: the translation of a Future Workshop* (2012). This last paper points out the use of social networks such as Facebook as a tool for Participatory design. This idea seems very in tune with what I see PD needs, as it gives the opportunity to reach millions of potential users from the commodity of their homes and attend the designers at their own convenience, reducing the resource use and adapting to the modern distributed workplaces.

If the PD community takes these two aspects into consideration and react, I believe that PD could become common practice in many software development companies. I do believe however, that PD is not suited for smaller companies and recent startups where specially designers and their time is a scarce resource. However if sufficiently adapted to take this into account along with aspects mentioned earlier it could also show some type of influence in practices of even the newest start-ups.

## 6.4.Example with Forecast.it

I will now proceed to describe the business model of Forecast.it and continue with a projection of how they wish their business to evolve and if that evolution would include the use of PD methodology.

Forecast.it is a software development company that develops software for enterprises. Their software helps companies estimate and execute projects saving money, and making the process of the project managing more clear, transparent and easier to follow. Their clients being normally large companies who carry out big projects. They offer their software through a platform which is common to all their clients. This platform can either be used online or downloaded into the clients computers. Along with their software they offer a training service for their software and a consulting service. In order to use the platform the client must pay a subscription fee annually. In the business model canvas you can see clearly how they conduct and manage their business.

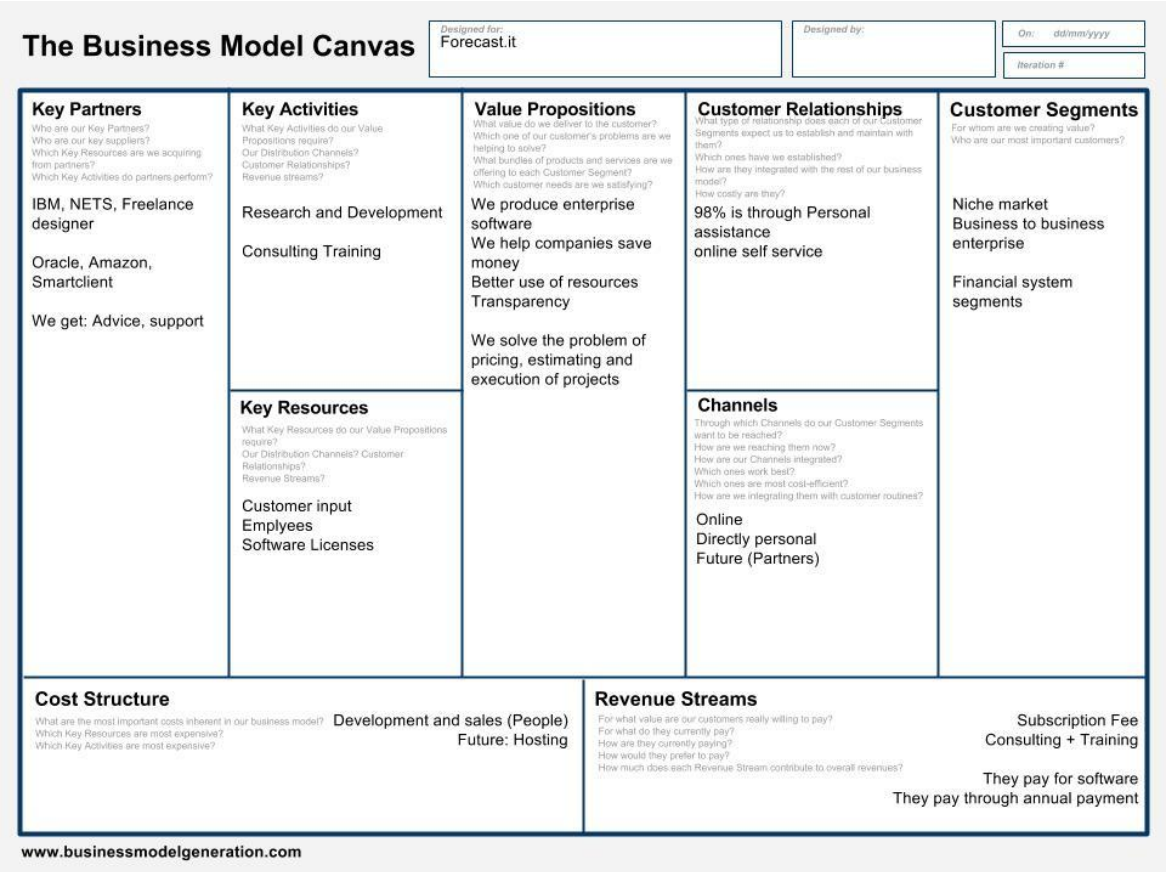



Figure 1 Forecast.it Business Model canvas

When clients subscribes they normally have specific requirements for the systems and needs which needs to solve, so clients normally negotiate with the company in order to have some software developed for the platform. So their development of software follows the scheme in Figure 2.

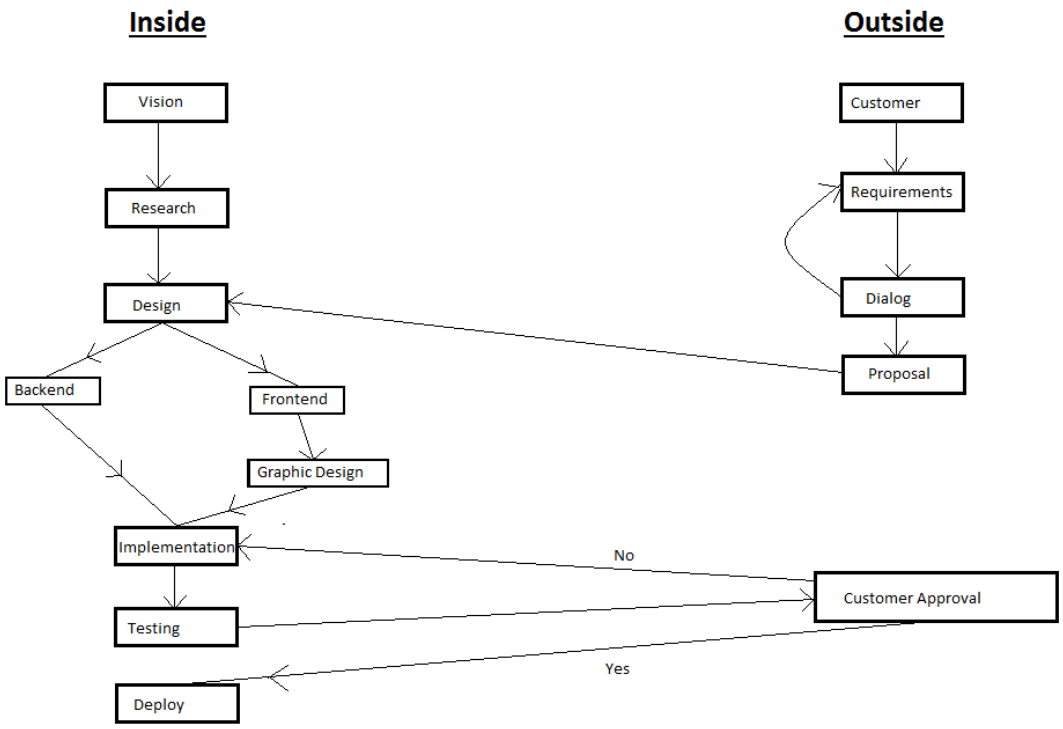



Figure 2 Forecast.it Development scheme

As we can see in the figure to develop the software development has two starts. Inside the company it starts with their own vision and research into what they wish to offer clients. Outside the company a customer normally states its requirements and proceeds to have a dialog with Forecast.it about their needs and problems and how they can be satisfied and resolved. It is in this stage where Forecast.it sometimes proposes better solutions to their clients than the ones they came up with by themselves. Once the requirements are set it is put on a proposal and the design process begins. The backend of the design process is then done by programmers and developers at Forecast.it. The front end is then sketched up by Forecast.it and handed to a freelance graphic designer to make the user interface. Once it is programmed in the backend and the graphic designer has finished his part it is implemented into the system and subsequently sent to the client to gain his approval. If the customer approves it is then deployed in the system, if the customer does not approve they have return to a dialog and when it is clear what is wrong and which changes are needed they return to the implementation phase.

In the future Forecast.it wishes to externalize their consulting and training services to other partners they will work with across the globe. This frees them up from those tasks, so they can concentrate entirely on the development of their software platform. So they would not have to deal with anything else besides development. So in a future if Forecast.it would wish to use PD in their business as it is today their development process would look something like Figure 3.

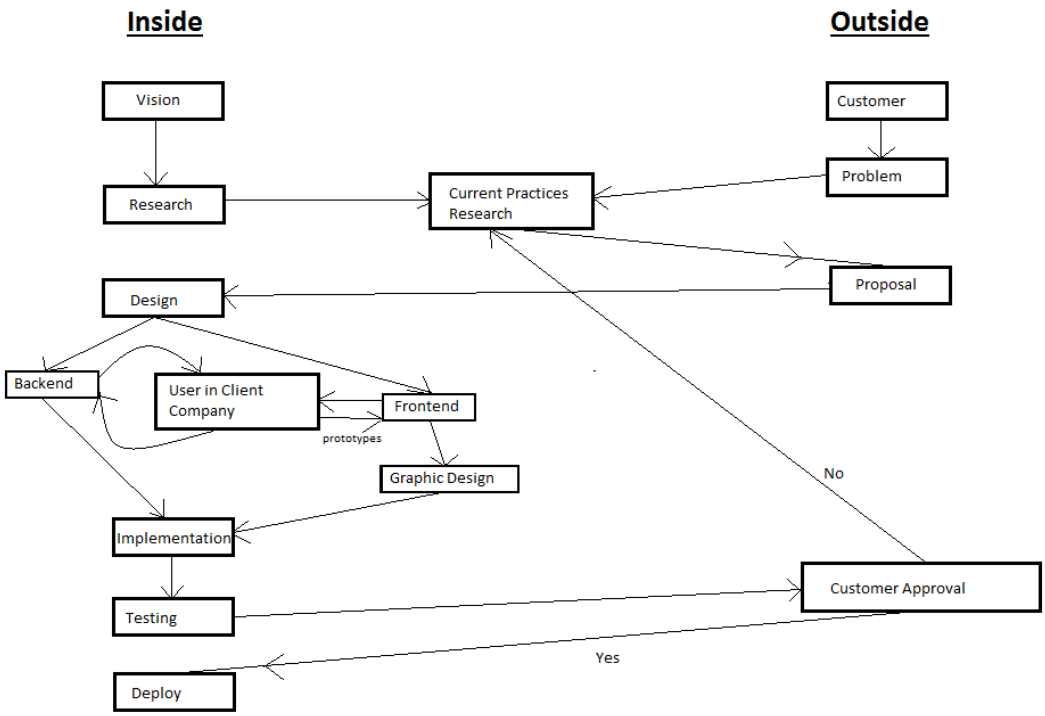



Figure 3 Forecast.it Development Scheme with PD

As we can observe there are quite a few changes. First of all although the company has a global vision of their platform and how they wish to develop it, they do not proceed without first taking into consideration and researching the current practices in their clients. When a customer contacts Forecast.it they would no longer state their requirements, they would state a problem they wish to solve. After stating a problem a research of current practices would go underway and have as an output a proposal for a solution with their platform much like they have now. Once the proposal is written and all parties agree, the design phase would start. The design phase would again be split in two: the backend and the front end. The backend would be designed first. In the backend the basic functionalities are designed. We can see that in this scheme there is a constant communication with a user of the system in the client company. Here the Forecast.it developer would with the help of the user design its functionality. In the frontend, which is worked on after the backend, the user would help with the design of the user interface. As it is a very popular PD technique prototypes would be used in this phase, be it paper mockups or more sophisticated prototypes. Once the user is satisfied with the prototypes and its future functionalities they would proceed to the implementation. Again after implementation testing would begin. In this case they must also gain the approval of the customer which is the companies approval. As it was designed with the help with one of the companies employees, the company would unlikely not gain the approval but if the rare case were to happen, this would mean that clearly the research of current practices was not thorough enough. So it would be necessary to return to this step in the scheme and continue from there, obviously using what has been worked on so far.

Another big difference in this case is the proposal. This time it would have to state an employee of the client company to partially work with Forecast.it. This of course would shift some aspects of their business model. Of course their revenues from consulting and training would be gone. Their inclusion of the employee in the design process might have to lower the price for the inclusion of their needs in the platform, but since the quality of the platform would increase, a higher subscription fee would be viable. Forecast.it plans to have clients from all over the world, so PD would not be possible unless methods are developed, as it has been discussed in the previous section of this chapter, that permit PD across long distances. So if even before Forecas.it thinks of applying PD they must tackle the issue of getting access to users. At this moment of PD development right now, if they were to have a worldwide network of clients their only option would be to produce users using local participants which they could use as users. The quality of these users would not be the same as those produced by a client company, but PD has experience obtaining users where there is none, such as users for new systems and the likes. Of course this introduction of PD has to be studied by Forecast.it if it is viable in their market sector and with their regular clients.

# 7. Conclusion

The interaction between Participatory Design and Business modeling is a fundamental issue for the future of the Participatory Design community. As such the research of this interaction is beginning to have more presence in the Participatory Design community as has been pointed out by Kyng (2010). So through the execution of this project of this project I have begun to research this relevant and complicated field. I have found that while Participatory Design and business modeling do share very relevant common interests, there is also strong clashing interests as well.

These clashing interest grow from the core visions the two fields have. While Participatory Design seeks to empower the user and give him voice in the creation of IT systems he will use, the more traditional capitalist businesses I have studied seek to make better products in order to gain an advantage over their competitors and thus gain wealth and profit. But even with this fundamental clash of visions there is still shared interest. Through my company study I found that software development companies constantly worry about obtaining information from their users. They seek to gain as much possible information from the user in order to improve the software adapting it to their real needs. Participatory Design has vast experience on this topic, as one the key activities of Participatory Design is finding out what users need and what are the real problems in the workplace. Other findings through my company study include the knowledge that most software developers for enterprise software have to include a vast manual in order to use their software and have to offer a training service as well in order to train workers of the client company. This demonstrates that the software development industry produces very complex and not user friendly software, leaving a large gap for improvement in the products and techniques of producing those products. This gap also leaves a door open for the introduction of Participatory Design. Other findings include

the common problem in Participatory Design of not finding proper users, or users not wishing to participate in the design process. This finding was common to the company Gluu which offered to give feedback on the software and vote for the design, but was largely unused. So as it has done with other IT design projects, Participatory Design must tackle this issue.

Through my Literature review of papers in the Participatory Design community such as Participatory Design In Economic Terms: A Theoretical Discussion (1998) I found arguments for Participatory Design being economically benefiting. But this paper does not include many impracticalities of Participatory Design such as its large use of resources and argues that is saves money and other resources in aspects such as training of future workforce for a company. But companies prefer to delay payments to a later time and would not wish to invest on software which in five to ten years might be replaced as it is the nature of software. Because of these impracticalities of large resource usage, it would seem logic that Participatory Design seems more adequate for bigger and well established companies or companies with specific business models such as customer relationship oriented businesses which have an emphasis on clients and users. As start ups and small companies have less resources and need to move at a faster pace to get entrance in the market.

So in order to further introduce Participatory Design in businesses it must adapt to each sector individually. Although there are common steps which must be taken by the Participatory Design community in order to further close this gap between the fields. They must first reduce the resources and impracticalities of Participatory Design. As this is one of that main drawbacks, and some user participation is better than none. Another step, which has already begun to take place, is the acknowledgement of new telecommunications technologies which have revolutionized the world in the recent decade. The use of these technologies have revolutionized many technology sectors and it could possibly revolutionize Participatory Design as well. In a more globalized world where live in today outsourcing of certain jobs is inevitable so users and designers might be separated by thousands of kilometers and thus not be possible to use techniques and tools face to face, and only through virtual means. This latter step could in itself help the first step take place, and thus adapt to the business reality.

Lastly with the example of Forecast.it we have seen how a business model of a well established software development company would look like in a future where they include Participatory Design in their approach to software development. This example helps visualize how a company can organize and work with a user involvement policy.

Though the partial company study has proved most useful and relevant for me in the execution of this project, I do regret not being able to work and study a company whose policies towards users included more user involvement. This would have helped me greatly contrast what made these companies and their business model profit from partaking in these techniques. Helping my study be more plural and with more dimensions than the more traditional software developing methods. I would have also

wished to include in my company study bigger and more sector established companies. I would have desired companies who were older  well establish companies that would have had to adapt to changes in the software sector in the past, in order to find out how they approach users and what they were willing to do to obtain maximum product quality.

This paper is aimed more at starting the step of adapting Participatory Design to businesses. This area of research needs more in depth studies then could possibly be covered in this project. Just like Kyng(2010) paper starts the debate concerning the gap between the project politics and the techniques developed by the Participatory Design Community this project intends to continue this discussion in a specific area. The Participatory Design community cannot afford to ignore such a vast sector and thus must research it in depth.

# References


1. Simonsen, J., Robertson, T. (2012). *Routledge International Handbook of Participatory Design*. Routledge.
2. Greenbaum, J., Kyng, M., (1991). *Design at work: Cooperative Design of Computer Systems*, Chichester, UK: Lawrence Erlbaum Associates.
3. Beyer, H. , Holtzblatt, K. (1997). *Contextual Design: Defining Customer-Centered Systems*. ElseVier.
4. Morten Kyng (2010), *Bridging the Gap Between Politics and Techniques: On the next practices of participatory design.* Scandinavian Journal of Information Systems, 2010, 22(1), 49-68
5. Bødker K., Kensing F., Simonsen J.,( 2004). *Participatory IT Design: Designing for Business and Workplace Realities*. MIT Press.
6. Ellen Balka (2010), *Bridging the Gap Between Politics and Techniques: A response to Kyng.* Scandinavian Journal of Information Systems, 2010, 22(1), 77-84
7. Timpka, T., Vimarlund, V., (1998). *Participatory Design In Economic Terms: A Theoretical Discussion.* Proceedings of the Participatory Design Conference, 1998.
8. Robertson, T., (1996). *Participatory Design and Participative Practices in Small Companies*. Proceedings of the Participatory Design Conference, 1996.
9. Buur, J., (2012). *Participatory Design of Business Models*. Proceedings of the Participatory Design Conference, 2012.
10. Osterwald, A and Pigneur, Y. (2009). *Business Model Generation: A Handbook for visionaires, Game Changers and Challengers*, Amsterdam: Modderman Druwerk, 2009.
11. Cusumano, M., (2008). *The Changing Software Business: Moving From Products to Services*. IEE Computer society.
12. Schief, M., (2014). *Business Models in the Software Industry: The Impact on Firm and M&A Performance*. Springer Gabler; 2014 edition.
13. Cooper, H., (1998). *Synthesizing Research: A Guide for Literature Reviews.* SAGE Publications.
14. Paul Luff, Jon Hindmarsh and Christian Heath, (2000). *Workplace Studies: Recovering Work Practice and Informing System Design.* Cambridge University Press.
15. Manten, A., (1973). *Scientific Literature Review*. Scientific Scholarly Publishing 5
16. S. K. Goyal, (1985). *Economic Order Quantity under Conditions of Permissible Delay in Payment.* The Journal of the Operational Research Society. Vol. 36, No. 4 (Apr., 1985), pp. 335-338
17. Molokken-Ostvold, K. & Furulund, K. M. (2007). *The relationship between customer collaboration and software project overruns*.
18. Apple Inc. (2004). *iOS Human Interface Guideline.* Copyright © 2014 Apple Inc.

19. Chris Jenks (1998). *Core Sociological Dichotomies*. London, England, UK; Thousand Oaks, California, USA; New Delhi, India: SAGE.
20. Grudin, J. (1993). *Obstacles to participatory design in large product development organizations*. Participatory design: principles and practices, 99-119.
21. Gumm, D. C., Janneck, M., & Finck, M. (2006, October). *Distributed participatory design–a case study*. In Proceedings of the DPD Workshop at NordiCHI (Vol. 2, No. 006).
22. Danielsson, K., Naghsh, A. M., Gumm, D., & Warr, A. (2008, April). *Distributed participatory design*. In CHI'08 extended abstracts on Human factors in computing systems (pp. 3953-3956). ACM.
23. Reyes, L. F. M. (2012). *Participatory Design through Social Media: the translation of a Future Workshop*.

# Appendix A

## Transcript of first meeting with Forecast.it

**Dennis Kayser**: If a customer asks us to build something really strange we obviously just build it for them, but most of the time we extrapolate it to something which is more useful for everybody so that its more generic. Most of the time when the customer has some strange request and we sit and think about it , we usually come up with a better solution that can actually fit everybody, instead of just doing some weird thing. We have some weird requests, and when we sat and drew it up in the board we thought: that is not what they want because that is not what they need. Then we kind of made a new solution and suggest it to them and usually turns out being the better thing to do. So we try to discourage people from doing to many weird things because most of the time it doesn´t work. I mean we have experience doing this, so it makes sense that we actually know more about this than they do.

**Niels Frederiksen**: Sometimes they think they know what they want and need, and then we convince them that it´s a bad idea, you should usually do it like this and this and it ends up being the better solution

**Dennis**: In the beginning we did it the other way around, and we just built whatever they asked for. And that was not good, so we ended up having to rewrite a whole lot of things because it just became a big mess.

...

...

...

**Interviewer:** How often or what percentage of failures do you have. in the sense that you get to the testing phase and the customer looks at it and says: "This isn´t what I wanted, this isn´t what I asked for".

**Dennis:** Rare, rare, very rare.

**Niels:** Yea, we haven´t had any like completely "no". We have had some "oh ok but could we add this, or tweak this a little bit". Actually for example at one instance we came to the customer approval meeting where we show off what we have, and then we kind of with the customer came up with a better solution. We talked a little about it: "Wouldn´t it actually make more sense if we changed this to do this, this and that". And it was a small change, so its something I implemented. We have never had a straght "no, this is not what we wanted so we won´t pay for it".

**Dennis:** We actually have a very close interaction on this, right. To make sure we actually get the right thing. But It´s just that the customer doesn´t have time for us to sit and call him every five seconds during this phase. It´s just going to take way too much time for us and too much time for them.

**Interviewer:** Ok, so you have not received complete no´s. But resource wise in time to correct errors and such?

**Niels:** There has been nothing lost, cause we have always been paid for the stuff that we do. I guess yes you can say that we have spent all together 10 hours of development, unpaid development, to change stuff.

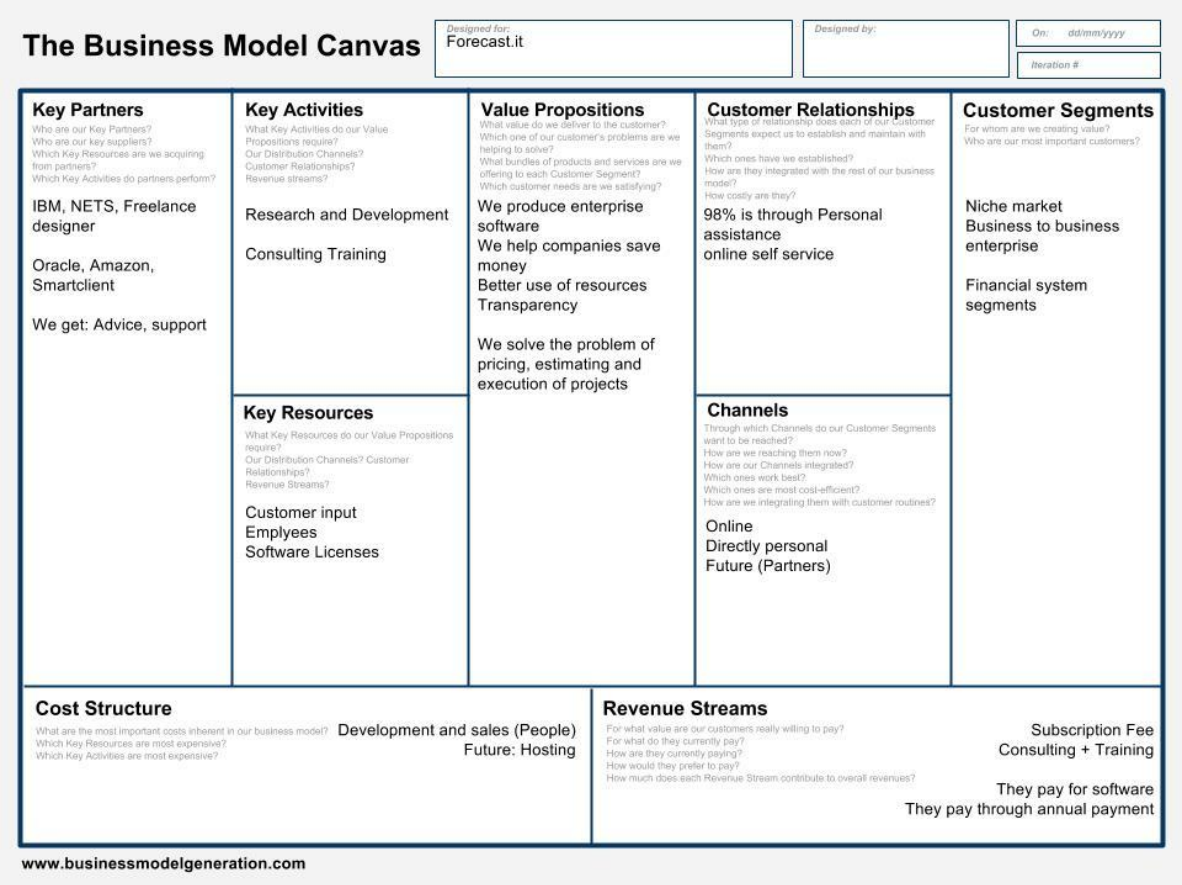
**Dennis:** its minuscule. And also when we do this we have a good price on this, so we can actually afford to most of the time do some stuff within the scope of the proposal. So we can still do some changes before it becomes a negative income for us.

## Transcript of first meeting with Gluu

**Interviewer:** Obviously you are still a start-up, but you say this can be global so it has a huge space for growth. Would you be open ro change the design process in order to gain more customer satisfaction, and maybe increase subscription fees, or would you still follow the same model as now?

**Soren:** No, we wouldn´t do that. We had it already designed it in such a way. The reason we are not designing with the customer, involving them more is that there are not enough of them to become involved. Sometimes it's not that problem that the customers cannot be involve, it's that the customers do not participate in the channels that are there. If you know what I mean. So we have a chance for people to vote on different functionalities, but we currently don´t get much input on it. This is why we need to interpret the best what users are doing and then create from there. It´s not a decision we made not to include them it's just a fact that they don´t get involved.

## Business Model of Forecast.it visualized through the Business model Canvas



**The Business Model Canvas**

Designed for: Forecast.it

Designed by:

On: dd/mm/yyyy

Iteration #

**Key Partners**

Who are our Key Partners?
Who are our key suppliers?
Which Key Resources are we acquiring from partners?
Which Key Activities do partners perform?

IBM, NETS, Freelance designer

Oracle, Amazon, Smartclient

We get: Advice, support

**Key Activities**

What Key Activities do our Value Propositions require?
Our Distribution Channels?
Customer Relationships?
Revenue streams?

Research and Development

Consulting Training

**Key Resources**

What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?

Customer input
Emplyees
Software Licenses

**Value Propositions**

What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?
What bundles of products and services are we offering to each Customer Segment?
Which customer needs are we satisfying?

We produce enterprise software
We help companies save money
Better use of resources
Transparency

We solve the problem of pricing, estimating and execution of projects

**Customer Relationships**

What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?

98% is through Personal assistance
online self service

**Channels**

Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?

Online
Directly personal
Future (Partners)

**Customer Segments**

For whom are we creating value?
Who are our most important customers?

Niche market
Business to business enterprise

Financial system segments

**Cost Structure**

What are the most important costs inherent in our business model?
Which Key Resources are most expensive?
Which Key Activities are most expensive?

Development and sales (People)
Future: Hosting

**Revenue Streams**

For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to overall revenues?

Subscription Fee
Consulting + Training

They pay for software
They pay through annual payment

www.businessmodelgeneration.com

Este documento esta firmado por

| | | |
|---|---|---|
|  | **Firmante** | CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES |
| | **Fecha/Hora** | Tue Nov 04 13:52:31 CET 2014 |
| | **Emisor del Certificado** | EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES |
| | **Numero de Serie** | 630 |
| | **Metodo** | urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature) |